논문 분석요약서

<table>
<tr><td>제출자 성명</td><td colspan="2">조송현</td><td>학번</td><td>2020511038</td><td>제출일자</td><td>2020. 5. 23.</td></tr>
<tr><td rowspan="4">논문정보</td><td>논문명</td><td colspan="5">Towards Automated Penetration Testing for Cloud Applications</td></tr>
<tr><td>저자명</td><td colspan="5">Valentina Casola, Alessandra De Benedictis, Massimiliano Rak, Umberto Villano</td></tr>
<tr><td>학술논문지명</td><td colspan="5">2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)</td></tr>
<tr><td>장표, 연번, 게재연도 등</td><td colspan="5">IEEE ISSN: 1524-4547. 2018</td></tr>
<tr><td colspan="7">논문 분석 요약</td></tr>
<tr><td>연구목적
(문제 제기, 연구 필요성 등)</td><td colspan="6">본 논문에서는 클라우드 보안 취약점 테스트 방법 중 하나인 Penetration Testing(Pentesting)을 발생시키고 실행하는데 도입할 수 있는 자동화방식을 소개한다. 저자는 Pentesting 환경설정 및 개시를 자동화하여 클라우드 애플리케이션에 대한 coarse-grained 수준의 평가를 가능하게 하는 프로세스를 소개한다.</td></tr>
<tr><td>관련 연구 현황</td><td colspan="6">Penetration Tests (Pentest) : 침투 시험
■ 클라우드 애플리케이션 발전은 관련 자원에 대한 제어 부재로 보안 문제를 대두시키고 있다. 침투시험(Penetration Testing)기술은 다른 적용 가능한 테스팅 방법들에 비해 상대적으로 강력하고 효과적이라 할 수 있다.
■ Penetration Testing은 그러나
1) 사람의 제어가 필요한 방식이어서 발생가능한 공격에 대한 이해(인지)가 동반되어야 하며
2) 테스트를 실행하는 데 사용될 해킹 도구에 대한 이해가 필요하며
3) 모든 상황에서 사용 가능한 방법은 아니라는 문제점이 존재한다.</td></tr>
<tr><td>접근방법
(문제해결 방안, 시험 절차 등)</td><td colspan="6">1. Penetration Testing Methodology(침투시험 방법론)
■ 이 논문에서 소개하는 pentest 방법론은 pentest 환경설정 및 개시를 자동화함으로써 노출된 취약점에 대한 coarse-grained 수준의 평가 및 클라우드 애플리케이션에 존재하는 위험 평가를 가능하게 한다.
■ 이 논문에서의 Pentest 방법론은 3가지 주요 단계를 제시한다: Preparation(준비); Scanning(스캐닝); Pentesting(펜테스트)
■ 각 단계는 모두 Model-based Activities (모델 기반 행동)과 System-based Activities(시스템 기반 행동)을 가진다. 모델기반은 카탈로그(Catalogue)에 기초하며, 모델링을 필요로 한다.

2. Preparation Phase (준비 단계)
■ Model-based Preparation phase: SuT risk analysis(모델기반 준비 단계: SuT 위험 분석):준비단계에서의 리스크 분석은 Multi-cloud Application</td></tr>
</table>

	Composition Model (MACM) 형식주의에 기반한 시험용 체계(SuT) 모델에서부터 시작된다. 각 구성요소 및 구성요소 간 주요 위협을 파악, 분류하고 위험순위를 정할 수 있도록 하는 자동화된 프로세스를 제공한다. ■ System-based Preparation phase: Testbed configuration (시스템 기반 준비 단계: 테스트베드 환경설정): 펜테스트 자동화 환경은 SuT 모델로부터 추출된 정보에 기초한 '사전 환경설정'을 이용함으로써 가능해진다. 테스트환경은 사전에 환경 설정된 네트워크(VPN)-3개 가상머신(공격 머신, 스캐닝 머신, 단말기 에뮬레이터 머신)을 연결한다-로 구성된 가상환경으로 조성된다. 3. Scanning Phase (스캐닝 단계) ■ 스캐닝 단계는 SuT에 영향을 미치는 약점 및 취약점을 확인한다. 이 논문의 접근에 따르면, 약점이란 모델기반 스캐닝에 의해, 카탈로그에 접근하는 방법으로 확인이 된다. 그러나 취약점은 스캐닝 도구를 이용, 취약점 스캐닝 과정을 수행하며 확인할 수 있으며 시스템기반 스캐닝의 결과물이라 볼 수 있다. ■ Model-based Scanning Phase: Weakness identification (모델기반 스캐닝 단계: 약점 확인): 모델기반 스캐닝은 단순 명료하다. 카탈로그를 이용, 약점을 구성요소에 대응하여 확인하기 때문이다. 따라서, SuT에 포함된 구성요소 타입에 따른 쿼리들을 이용, 약점을 쉽게 발견할 수 있다. ■ System-Based Scanning phase: Vulnerability Scanning(시스템기반 스캐닝 단계: 취약점 스캐닝): 취약점은 관련 technological stack 및 시스템 상 특정 소프트웨어와 관계가 있다. 알려진 취약점을 확인하기 위해서는 OpenVAS 도구를 사용한다. 4. Pentesting Phase (펜테스팅 단계) ■ 펜테스팅 단계는 적합한 공격을 구성(building) 및 실행(executing)하고 이전 단계에서 찾아낸 약점 및 취약점을 이용하는데 초점을 맞춘다. ■ Model-based Pentesting Phase: Preparation of the attacks (모델기반 펜테스팅 단계: 공격 준비): 모델기반 행동에서는 카탈로그로부터 알 수 있는 이용(exploits)과 공격(attacks)을 적절하게 조합한 공격을 준비한다. ■ System-Based Pentesting Phase: Execution of the attacks(시스템 기반 펜테스팅 단계: 공격 실행): 시스템 기반 행동은 사전에 정해진 순서에 따라 공격을 실행할 뿐이다. 각각의 공격 이후 시스템은 리셋되고 새로운 공격이 실행된다.
결론	■ 기존의 펜테스팅과 비교하여 본 논문에서 소개한 프로세스는 완전한 자동화를 이룰 수 있으며 넓은 약점 및 취약점에 광범위하게 대응할 수 있다.
향후 연구사항 (추가적인 연구분야, 연구내용 등)	■ 본 논문에서 소개한 방안에서는 '카탈로그'가 중요하므로 새로운 정보를 수집하는 것과 기존 정보간 연관성을 찾는 연구를 지속해야 할 것이다. ■ 프로세스 중 공격을 실행하는 방법을 더욱 개선하여 본 연구의 효용성을 더욱 높일 수 있어야 하겠다.